KAWASAKI STEEL TECHNICAL REPORT
No.20 (June 1989)
Information Systems

Techniques for Improving Productivity of System Development

Yukihiro Hiramatsu, Kenzo Ohsugi, Tohru Ariki, Masanori Minagawa, Masakuni Furukawa

Synopsis :
Kawasaki Steel has developed techniques for improving the productivity of system development in order to accomplish refurbishment of its total systems. The refurbishing techniques are centered on the data-oriented information resource management concept and segmentation and reuse of software. In view of the above, Kawasaki Steel has; (1) standardized information system development procedures and established data management structures, (2) developed a data dictionary system, (3) developed a system development and maintenance support system ranging from designing to production and maintenance, and (4) through the use of these techniques, doubled the productivity of system development and realized information resource management and utilization of computer-stored data by end-users.

The body can be viewed from the next page.

KAWASAKI STEEL TECHNICAL REPORT No. 20 June 1989

Techniques for Improving Productivity of System Development*


Yukihiro Hiramatsu
Senior Manager, Osaka Sales Office, Systems Engineering Dept. I, Systems Engineering Div. I, Kawasaki Steel Systems R & D Corp.


Kenzo Ohsugi
General Manager, Systems Planning & Data Processing Dept., Head Office


Tohru Ariki
Assistant General Manager, Mizushima Office, Systems Engineering Div. II, Kawasaki Steel Systems R & D Corp.


Masanori Minagawa
Assistant General Manager, Kobe Office, Systems Engineering Div. II, Kawasaki Steel Systems R & D Corp.


Masakuni Furukawa
Senior Manager, Kobe Office, Systems Engineering Div. II, Kawasaki Steel Systems R & D Corp.

Synopsis:

Kawasaki Steel has developed techniques for improving the productivity of system development in order to accomplish refurbishment of its total systems. The refurbishing techniques are centered on the data-oriented information resource management concept and segmentation and reuse of software. In view of the above, Kawasaki Steel has: (1) standardized information system development procedures and established data management structures, (2) developed a data dictionary system, (3) developed a system development and maintenance support system ranging from designing to production and maintenance, and (4) through the use of these techniques, doubled the productivity of system development and realized information resource management and utilization of computer-stored data by end-users.

1 Introduction

At Kawasaki Steel, an age of refurbishment began about 1980 following the completion of plant construction and computerization of information systems. With customer requirements, including product specifications, quality, and delivery dates becoming both more diverse and increasingly rigorous, it became necessary to upgrade production methods and refurbish production facilities and information systems in order to improve price and non-price competitiveness[1].

Since the early 80s, information systems have undergone major qualitative and quantitative changes. In parallel with production systems trends toward process integration and synchronization, information systems have also been refurbished to realize a closer linkage between systems and to accomplish systems integration.

To carry out this refurbishment, techniques for improving the productivity of systems development were developed and applied, making use of an approach based on information resource management (IRM), in which methodology, structures, and tools are incorporated in a single unified concept.

A common methodology and structure were shared in all divisions of the company. The company employs two main groups of tools; the ones available in a FACOM environment common to all company works, and the others, available in an IBM environment, used in the head office.

This paper describes in outline these techniques and their effects, and briefly discusses future tasks. Incidentally, FACOM-based tools are mainly dealt with.

* Originally published in *Kawasaki Steel Giho*, **20**(1988)2, pp. 151-157

2 Concept of Improvement of Productivity of Systems Development

2.1 Background

The development of information systems is conducted by a project team representing both the user department and the computer systems department. In the past, the per-project scale of software was about a maximum 500 000 COBOL steps. After 1980, however, this figure increased to one to three million steps. In addition, interfaces between systems via data bases and communication lines became complex owing to system integration. On the other hand, because systems refurbishment had to be synchronous with the renovation of production facilities, it was necessary to carry out speedy development with limited personnel.

2.2 Problems

The following items were selected as problems to be solved:

(1) A data base management system (DBMS) has been introduced, and various means of data storage and access, data compatibility and security, and an improvement in the independence of data and programs have been realized. A system for obtaining an accurate understanding of each data item stored was now to be established in the construction of a true common data base and the realization of system integration and effective utilization of accumulated data on the basis of this common data base.

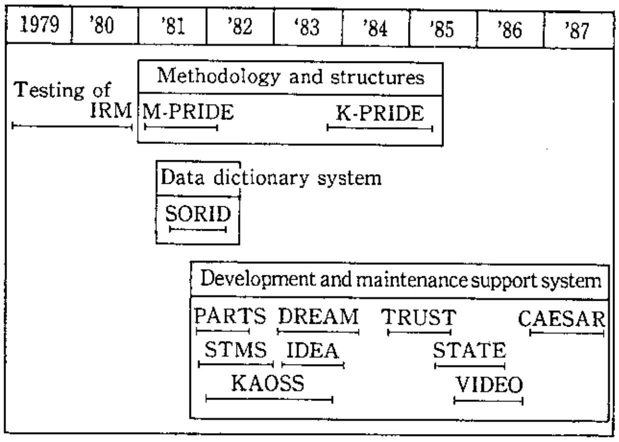
(2) Methodology for design in the upstream phase is to be established so that user department personnel and the computer systems department, in close cooperation, can efficiently carry out the design of large-scale systems in response to changes in production systems.

(3) Any substantial improvement in systems development productivity depends on the promotion of computer- and automation-supported work to replace manual work. In addition, the support system must cover a wide range of functions, from development to maintenance, in an integrated manner.

(4) The utilization of accumulated data by end users is to be expanded to reduce the amount of software development required of the systems department and to ensure rapid access to appropriate information.

2.3 Concept of Solution to Problems

As shown in **Fig. 1**, the development of techniques for solving these problems was carried forward in the order: (1) establishment of the development methodology and data management structures, (2) development of a data dictionary system, (3) gradual development of development and maintenance support systems.



M-PRIDE: Mizushima PRIDE
K-PRIDE: KSC PRIDE
SORID: System organization and resource information dictionary
PARTS: Pattern and parts oriented application requirement translation system
STMS: Standard table management system
KAOSS: KSC all-round online support system
DREAM: Database retrieval system for effective analysis and maintenance
IDEA: Information design assistant system
TRUST: Toolsware for reliability up of software testing
STATE: System traffic analysis dictionary under temporary execution environment
VIDEO: Visual image design aid by end-user operation
CAESAR: Computer aided engineering for software automation and re-use

Fig. 1 System development history of Kawasaki Steel

2.3.1 Basic concept

IRM was the key concept for an overall solution to these problems. Kawasaki Steel, introduced IRM on a trial basis using a method of its own devising since 1979. IRM considers data, which is the raw material of information, as a vital corporate resource, on par with the traditional three basic resources, *hito, mono,* and *kane* (people, things, money).

The management of data requires that data items be established unequivocally through clear defining attributes of data items shown in **Table 1**. The content of such definitions includes data name, number of digits, and units. Proper management of data is essential to the compatibility of systems and the effective utilization of accumulated data.

2.3.2 Establishment of development methodology and data management structures

To realize IRM, it is necessary to adopt a design method in which attention is paid to information and data, rather than a conventional method centered on functions and programs only. This design method is called the data-oriented approach. Kawasaki Steel intro-

Table 1 Typical attributes of data item

1. Data name
2. Entity class
3. Type
4. Number of digits
5. Definition/application
6. Origin
7. Administrating department
8. Check Content
9. Method of generation
10. Related programs
11. Related input and output information
12. Related records and files

duced PRIDE (Profitable Information by Design-through Phased Planning and Control), a methodology for information systems development developed by the American firm MBA, and established its own standards for systems development work on the basis of this methodology.

At the same time, a data management structure was established in which data managers served as key personnel with coordination responsibilities.

2.3.3 Development of data dictionary system

The number of data items frequently runs into the tens of thousands, making strictly human management impossible. A viable IRM system must, therefore, reduce the total work load, and not merely minimize increases in the load due to data management. A data dictionary system supports this. Since there was no commercial data dictionary system that permitted the use of Japanese ideograms at the time of IRM introduction, it was necessary for Kawasaki Steel to develop its own system.

2.3.4 Step-by-step development of development and maintenance support systems

The data dictionary system provides the basis for the development and maintenance support systems, and therefore was developed first; peripheral systems were then developed on the basis of the data dictionary system. Eventually it was possible to build an integrated development and maintenance supprt system with a good overall balance.

3 Measures to Improve Productivity of System Development

3.1 Establishment of Methodology and Data Management Structures

3.1.1 Establishment of methodology[2,3]

After the introduction of PRIDE, standardization activities proceeded to ensure a smooth application of PRIDE to largescale projects. As a result, M-PRIDE (Mizushima PRIDE) was established, followed by K-PRIDE (KSC PRIDE) for company-wide application.

The following improvements were made in the course of standardization:

(1) Types and forms of documents, development work items, and work assignments were established on the basis of PRIDE.

(2) A data base design procedure corresponding to the system design procedure was established. The data base design procedure is shown in **Table 2**. Through this procedure, a process of conceptual data base design was established, with entity type analysis at its core. The aim of this work, however, was not mere analysis, but design work as such. Both the entity types which were the result of this process and their relationships were important as: ① a framework for organizing a large number of categorized data items, ② a starting point for data base design, and ③ a means of testing system design.

(3) Standards for various data management activities,

Table 2 K-PRIDE database design process

Phase	Activity	Tasks
Phase 1 (Conceptual system design)	Activity h (Conceptual database design)	1. Identification of control items 2. Definition of entity classes 3. Determination of entity class relationships 4. Grouping of entity classes 5. Description of conceptual database specifications 6. Walkthrough of conceptual database specifications
Phase 2 (Basic system design)	Activity f (Internal database design)	1. Review of conceptual database specifications 2. Establishment of internal database 3. Description of internal database relationships 4. Design of internal database structure 5. Description of internal database specifications 6. Walkthrough of internal database specifications
Phase 3 (Detailed sub-system design)	Activity e (Database design for DBMS)	1. Review of internal database specifications 2. Design of logical structure for DBMS 3. Design of record layouts 4. Design of housing structure for DBMS 5. Walkthrough of database structure for DBMS 6. Study of physical structure for DBMS 7. Study of database shift 8. Description of database specifications 9. Walkthrough of database specifications

```
 1  JIS COBOL  V10L11   BU7111          日付 88.01.13  時刻 09.37.01       制作 SORIDシステム           16 頁

SEQNO  A   B                                                                DD-NO    日 本 語 名 称
PROCEDURE DIVISION USING FCOM KCOM SCOM DBIRCB IN-MSG
                         IN-MSG1 BUB01R01 BUB01R02
                         BUB02R01.
**************************************************************
**   #1  (P.3)                                              *
**     MAIN                                                 *
**************************************************************
MAIN  SECTION.
MAIN-START.
    PERFORM  REC-HENSHUU-SHORI.
    PERFORM  SOUGYOU-REC-HENSHUU-SHORI.
MAIN-END.
    GOBACK.

**************************************************************
**   #1-1  (P.4)                                            *
**     REC-HENSHUU-SHORI                                    *
**************************************************************
REC-HENSHUU-SHORI  SECTION.
SRIN.
    IF  DD-BVC69 OF IN-MSG  =  SPACE                                        BVC69    中修理回数
        CONTINUE
    ELSE
    IF  DD-BVC69 OF IN-MSG  >  '5'                               861218A   BVC69    中修理回数
        CONTINUE   .                                             861218A
    ELSE                                                         861218A
        IF  DD-BVC35 OF IN-MSG  =  SPACE                                   BVC35    修理完了年月日
            CONTINUE
        ELSE
            MOVE  DD-BVC69 OF IN-MSG1  TO  DD-BVC69 OF BUB01R01.           BVC69    中修理回数
    IF  DD-BVC14 OF IN-MSG  =  SPACE                                        BVC14    冷間ガラ落し回数
        CONTINUE
    ELSE
        MOVE  DD-BVC14 OF IN-MSG1  TO  DD-BVC14 OF BUB01R01.               BVC14    冷間ガラ落し回数
    IF  DD-BVC42 OF IN-MSG  =  SPACE                                        BVC42    修理後受銑回数
        CONTINUE
    ELSE
        IF  DD-BVC35 OF IN-MSG  =  SPACE                                   BVC35    修理完了年月日
            MOVE  DD-BVC42 OF IN-MSG1  TO  DD-BVC70 OF BUB01R01            BVC42    修理後受銑回数
                                                                            BVC70    中修理後受銑回数
        ELSE
            MOVE  ZERO                 TO  DD-BVC70 OF BUB01R01.           BVC70    中修理後受銑回数
    IF  DD-BVA32 OF IN-MSG  =  '    .  '                                   BVA32    トピード秤銑値 (炉代始)
        CONTINUE
    ELSE
        MOVE  DD-BVA32 OF IN-MSG1  TO  DD-BVA33 OF BUB01R01.               BVA32    トピード秤銑値 (炉代始)
                                                                            BVA33    トピード秤銑値 (中修理後)
SREXIT.  EXIT.
```

Fig. 2 An example of COBOL compile list (Japanese data names)

including rules for naming data items, were established. A method by which terms are standardized and data items named by combinations of these terms was especially effective. These terms are called "phrases" and are also used in the retrieval of data items.
In our database design, control of data items means control of attributes rather than of domains (definition area). Five-digit codes are used as keys for identifying data items. The code is composed of an entity type number and a serial number within the entity type.
The integrated use of this number from documents to programs was effective in the unequivocal identification of data. As shown in **Fig. 2**, interpretability was improved by automatically adding data item names (in ideogram form) to a program list.

(4) A procedure for establishing data communication, corresponding to the system design procedure, was also established.

3.1.2 Establishment of data management system[3,4]

A data management system was established during standardization, as shown in **Fig. 3**; data management responsibilities are listed in **Table 3**. A data manager is responsible for and coordinates each project. In addition, a data management office was established for each place of work, with integrated control at each location aimed at. However, phrases and data item names related to order receiving are controlled in integrated fashion on a company-wide basis.

3.2 Improvement of Development Support Tools

Development and maintenance support system groups, including SORID, cover a wide range of functions while maintaining mutual linkage, and form a

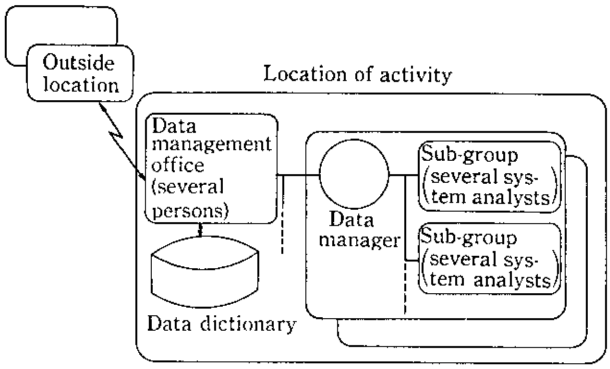


Fig. 3 Data management system

Table 3 Data management responsibilities and functions

Level of responsibility	Functions
Data management office	1. Intra-company normalization of data 2. Normalization of data among projects 3. Definition of new terms to be registered 4. Study of data management system 5. Standardization of data management activities 6. Dissemination of data management information
Data manager	1. Overview of data administration by project 2. Entry of new data items in data dictionary 3. Determination of new terms to be registered 4. User contact
System analyst	1. Determination of data names 2. Determination of data descriptions 3. Actual use of data dictionary system

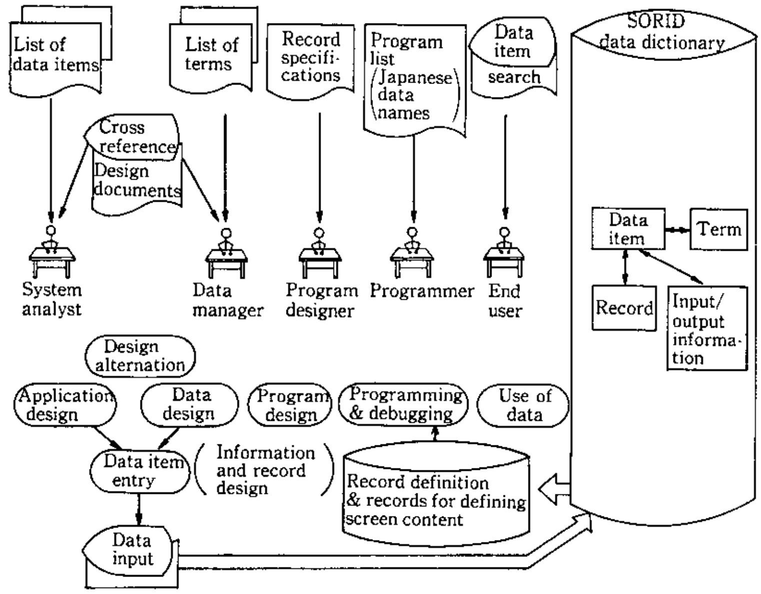


Fig. 4 Relationship between functions of SORID and system development process

comprehensive integrated development and maintenance support system. The operating system for these system groups is the Fujitsu, Ltd. MSP (Multidimensional System Product).

An outline of the systems is given below in the order of support phase.

3.2.1 SORID data dictionary system[1,2,3,5]

The relationship between the functions of SORID and system development work is illustrated in **Fig.** 4. In the basic system design phase, data item attributes are registered in the data dictionary after data items have been established and the data descriptions entered. The design of input/output information and records is supported by the data dictionary, and results are also stored. Data dictionary functions include document output, cross-referencing, and quick search for data items in response to commands involving combinations of phrases.

3.2.2 VIDEO screen design support system[3,5]

An outline of the VIDEO system is shown in **Fig. 5**. With this system, application system screen designs can be created on the terminal screen using actual screen images. Screen definitions and screen designs are automatically generated when screen definition informa-

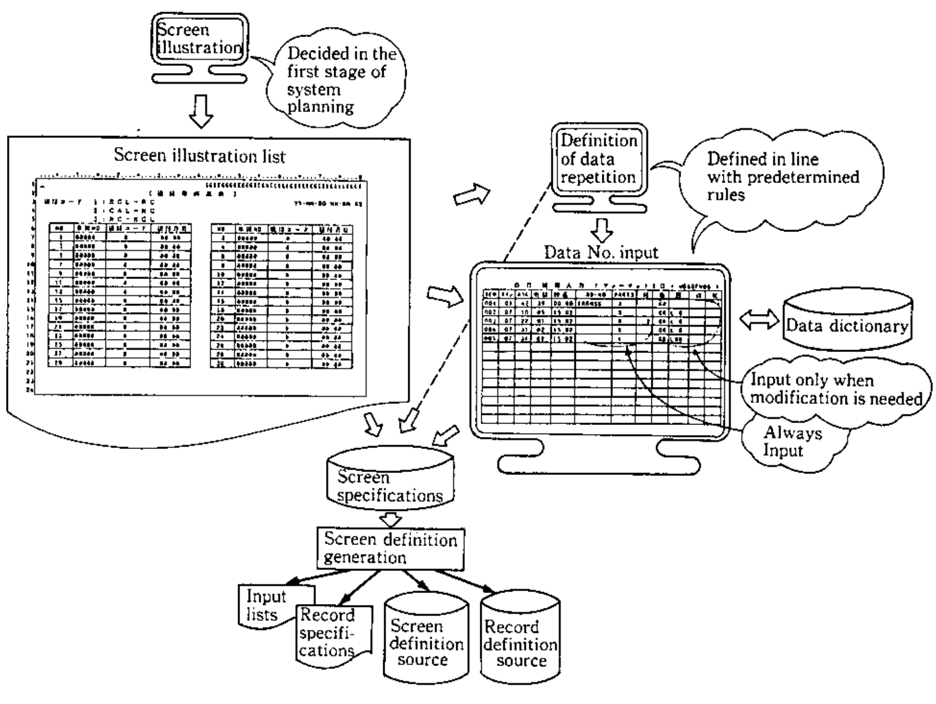


Fig. 5 Sketch of VIDEO system

tion concerning the screen being designed is entered. Data item attributes are available in the data dictionary; design results are also stored in the data dictionary. With VIDEO, it was possible to reduce screen design man-hours by more than 80%, and for end users to design their own screens.

3.2.3 STMS (standard table management system)

In this system, standard tables for specification retrieval used in the process for determining manufacturing specifications, etc. are separated from programs and are controlled centrally. When standards and specifications are revised, data must be immediately corrected, so that line operators also responsible for data maintenance carry out corrections at terminals without the help of system experts. When corrected results are approved, input data are converted to an efficient table form and are then updated during the operation of the system. Furthermore, a subroutine, used when a program retrieves a table, is also provided.

Although this system was previously in use, it was substantially improved on this occasion, mainly in the data correction and data update methods.

3.2.4 KAOSS on-line support system[6]

The KAOSS system works with AIM (Advanced Information Manager)/ACS (Application Control System), a Fujitsu Ltd. data base and data communication package, which it supplements.

The functions of this system include communication between application programs, including between different types of computers, distribution of on-line documents, and automatic start and stop of on-line systems. It also compiles various types of log information, including the contents of input and output messages.

3.2.5 PARTS program generator[3,5,7]

This system automatically generate COBOL sources concerning main on-line program and input data check modules. In on-line conversational processing, a series of conversations constitutes a function. The flow of this conversation is expressed in patterns, and main on-line programs modules are automatically generated. The check conditions for data validity are defined beforehand for each data item and are stored, and selection and decision table methods are used as means of defining check conditions. Input check modules are automatically generated on the basis of such definitions and the definitions of input and output information in the data dictionary.

About 42% of on-line programs are generated automatically.

3.2.6 CAESAR full-range program generator

CAESAR is an advanced program generator which complements PARTS by covering all areas not covered by PARTS. All COBOL source programs are automatically generated by the segmentation and reuse of software. This system was developed jointly by Kawasaki Steel and Fujitsu, Ltd.

Table 4 CAESAR system structure

1. Reusable component drafting system	1. COBOL-type component definition 2. 2-dimensional table-type component definition 3. Decision-table-type component definition 4. Data editing component definition
2. Program synthesis system	1. Conversational program design 2. Program generator 3. Program document generator 4. Application document generator

The functional structure of CAESAR is shown in **Table** 4. The features of the system are described in the following.

(1) The definition of components is conducted at a work station. Various means of definition are provided and, at the same time, a man-machine interface of the multiple window input-guidance type is realized. The data dictionary used for the definition of components is loaded down from the host computer, and completed components are loaded up to the host computer.

(2) Components are centrally controlled by the host computer to ensure full-range availability. The host computer synthesizes components and generates programs.

(3) Components are assembled in a man-machine interactive system in which input is guided by the menu selection method. With guidance from the computer, the operator follow through a menu tree from upper classes to lower selecting components in each menu and deciding variables when requested. On completing a tree of menus, COBOL programs are automatically generated by batch processing.

(4) All programs are generated by assembling components. Programs thus generated are complete. In other words, no manual correction of programs after generation is required.

(5) By managing component versions, the sharing of components for development and maintenance among multiple projects is possible.

With the aid of general-purpose components prepared in-house, about 80% of programs can be generated from reusable components. The remaining 20% are functions (components) peculiar to individual application programs.

Application of this system resulted in a reduction of about 65% in program design and programming phase man-hours, or a three-fold increase in productivity. When the effect in other phases is taken into account, an overall reduction in man-hours of slightly over 30%, or a 1.5 times improvement in productivity, was achieved.

3.2.7 TRUST test support system[3,5]

This system serves to support:
(1) Test data preparation
(2) Test execution
(3) Test result verification
The data dictionary is used in functions (1) and (3). The use of this system has increased efficiency in the test phase by about 26%.

3.2.8 DREAM general-purpose retrieval and renewal system for AIM data base[3,5]

In this system, access to data is in accordance with the data base structure; the access path is called the access pattern and is defined beforehand. It is possible to retrieve and update any item by specifying the access pattern, retrieval data item, and object conditions. It is also possible to enter retrieval requirements in their entirety beforehand. In this case, only the name of the retrieval item and the value of the retrieval key need be input. This system is effective in the test and maintenance phases and allows direct use of data by end users.

Furthermore, a subroutine that enables an application program to access the data base easily is provided.

3.2.9 STATE system dictionary system[3,5]

In the STATE system, it is possible, on the basis of data items, records, input and output information, etc., to grasp in real time the programs which process such items, as well as the contents of the commands in the programs. It can also be understood whether the data item and record in question are updated ones by the program or material for reference use only.

This system is effective in examining the causes of abnormalities and the scope of the effects of corrections in the test and maintenance phases. Significant effects can be expected when not only tools but also data items are clearly established.

3.2.10 IDEA general-purpose data analysis support system[3,5,8,9]

End users can directly analyze result data on quality and steelworks operation and can obtain necessary information rapidly. Necessary data is selected from the result data for each application system and stored and updated. Analysis work is also supported by TSS screen guidance.

FOCUS (sold by Assist Co., Ltd.), EASYTRIEVE (sold by Pansofic Co., Ltd.), and PLANNER (produced by Fujitsu, Ltd.) are used as end-user languages. All definitions required by these languages are automatically generated from the data dictionary.

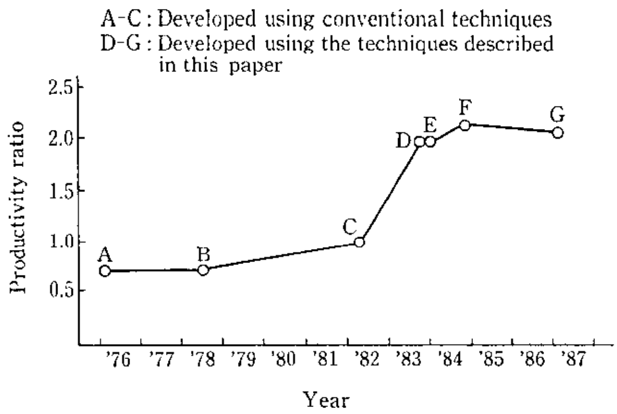


Fig. 6 Productivity of major systems projects at Mizushima Works

4 Effects

4.1 Improvement in Development Productivity

Development productivity trends at Mizushima Works are shown in **Fig. 6**. Productivity approximately doubled following this technical development. This value, it should be noted, includes not only the effect of the techniques described in this paper, but also the results of efforts of persons in charge of development in connection with refurbishment and of many other personnel development activities. The refurbishment plan, initially considered difficult, was completed successfully owing to this improvement in productivity.

4.2 Realization of IRM

At present, about 40 000 data items are registered in the data dictionary, and data descriptions for each data item have been prepared. It was thus possible to construct a common data base in which this data is stored, and develop a system of unprecedented scale. It may be said, therefore, that IRM was realized in both name and reality.

Unequivocality of data items ought to be verified by human intelligence and a level of more than 90% may be claimed. In IRM, however, no final completion is possible. It is, therefore, necessary to persevere in efforts to refine and improve the system.

Although IRM has long been a subject of study and development, it has rarely been realized, and the success of Kawasaki Steel's efforts in this area is very significant for the company's future information systems development.

4.3 Expanded Use of Accumulated Data by End Users

At present, Mizushima Works has about 25 gigabytes of data accumulated for analysis, and end users have developed analysis programs totalling about 500 000 steps. This volume of programs is equivalent to about 5 million steps in COBOL, which means that the equivalent of several large-scale systems has been developed by end users.

Thus, the program development burden on the systems department has been substantially reduced, and an improvement in the productivity of software development by users realized, while, needless to say, rapid and appropriate access to information has contributed to work quality improvement in user departments.

5 Future Problems

The following problems connected with the improvement of the productivity in systems development have been targetted.

(1) Improvement of the Functions of the Integrated Systems Development and Maintenance Support System
A productivity management system provided with a scale estimation model and a productivity model is to be introduced or developed in order to improve estimation accuracy in the planning stage. A project management system is to be developed on the basis of this productivity management system.

(2) Introduction of 4GL in Systems Department
In recent years, many fourth generation languages (4GL) have been marketed for use by systems departments in developing application systems in an on-line data base environment. These languages are to be examined. The optimum use of COBOL source program generators and 4GLs will be a goal.

(3) Establishment of Development Techniques for Distributed Systems Involving AI and Work Stations
The company has established development techniques using large business computers. The scope of application of distributed systems involving AI (artificial intelligence) and work stations seems likely to expand in the future. Development techniques are to be established in these fields through actual application.

6 Conclusions

Techniques developed by Kawasaki Steel for improving the productivity of systems development have been described. The results are as follows:

(1) In establishing systems development structures, the PRIDE methodology for information system development was introduced. PRIDE was improved in line with the specific requirements of the company and standardized. At the same time, a data management system was established for integrated data management at each place of business.
(2) The SORID data dictionary system using Japanese ideograms was developed to realize the IRM information resource management system.
(3) A comprehensive development and maintenance support system covering all phases, including design, manufacture, and maintenance, was developed.
(4) By applying these systems, it was possible to approximately double the productivity of systems development, conduct improved information resource management, and make accumulated data available to end users.

These techniques were developed using an approach in which methodology, structures, and tools were incorporated in a unified process, and are based on IRM, an approach of data-oriented design and the segmentation and reuse of software.

Although integrated system development and maintenance support systems began appearing on the market between 1983 and 1985, Kawasaki Steel had been active in this area since 1981. Therefore, it may fairly be said that the company's activities in realizing IRM and building an integrated systems development and maintenance support system were pioneering ones.

As software engineering continues to develop in the future, the authors believe their approach will have universal validity and intend to make further improvements on the basis of the present techniques. It will be necessary to realize the compatible use of and commercial tools such as the Σ system.

The authors would like to extend their sincere thanks to many people concerned, especially at Fujitsu, Ltd. and IBM Japan, Ltd., for their support in the development of these techniques for improving productivity.

References

1) S. Baba: *FACOM Journal*, **11**(1985)5, 6-15
2) Y. Hiramatsu, S. Baba, and S. Kinoshita: *FACOM EDP Ronbunshu*, 16(1983), 1-15
3) N. Ishizuka, T. Ariki, and N. Shiwaku: *Jimukanri*, **25**(1986)2, 67-75
4) S. Kinoshita: *IBM Review*, **95**(1985), 133-146
5) M. Tomisawa, T. Ariki, and N. Shiwaku: *FACOM EDP Ronbunshu*, 20(1987), 1-25
6) Y. Nagaoka: *FACOM EDP Ronbunshu*, 17(1984), 195-215
7) N. Shiwaku: Industrial Engineering for Iron & Steel Industry, **3**(1983), 52-59
8) M. Igata, S. Tanaka, and Y. Yamashita: *Fujitsu*, **37**(1986)3, 250-255
9) S. Tanaka, T. Ariki, M. Igata, S. Yamasaki, I. Takahashi, and T. Sekio: *Kawasaki Steel Giho*, **20**(1988)2, 138-143